LANGUAGE IN INDIA

Strength for Today and Bright Hope for Tomorrow

Volume 3 : 4 April 2003

Editor: M. S. Thirumalai, Ph.D.
Associate Editors: B. Mallikarjun, Ph.D.
Sam Mohanlal, Ph.D.
B. A. Sharada, Ph.D.

COMPUTATIONAL MORPHOLOGY OF TAMIL VERBAL COMPLEX

S. Rajendran, Ph.D.
S.Viswanathan, &
Ramesh Kumar

---

1. Introduction

It is a pan-Indian feature found in many Indian languages (such as Tamil, Malayalam, Kannada, Telugu, Hindi etc.) that their verb phrase consists of different combinations of verbs culminating into compound verbs which contribute to the interpretation of tense, mood, aspect, attitude, non-attitude, voice, causation, and negative polarity in a sentence. The structure of verbal complex (which include simple and compound verbs) is unique and capturing this complexity in a machine analyzable and generatable format is a challenging job. What is workable in the forward direction may not work in the backward direction. Building such a tool that analyzes and generates the complex verbal system with the proper interpretation of meaning involves skillful linguistic analysis and computational programming. Though the formation of the verbal complex is mainly anchored on the morphological level, it is linked to syntax on the one side and semantics on the other side. The formation of the verbal complex involves arrangement of the verbal units and the interpretation of their combinatory meaning. Phonology also plays its part in the formation of verbal complex in

termsof morphophonemic or sandhi rules which account for the shape changes due to inflection and juxtaposition.

The following are the major issues to be faced:

- What are the morphophonemic or sandhi rules which account for the shape changes in the formation of verbal complex?
- How the verbs are combined and how the meanings of the verbal complex are interpreted based on the combination of meanings of the constituents of the verbal complex?
- What are the possible and impossible permutations?
- How to restrict the generation of ungrammatical verbal complex and unlawful meaning interpretation?

Morphological analysis (parsing or recognition) is concerned with retrieving the structure, the syntactic and morphological properties, or the meaning of a morphologically complex word. Morphological generation is concerned with producing an appropriate morphologically complex word from some set of semantic or syntactic features. We need to understand the following things:

1. What kinds of representations a morphological processing system must deal in before we can appreciate the issues involved in designing such a system, or give a fair evaluation of the performance of a particular morphological processing system?
2. What sort of things morphology exhibit in a language?
3. How are words built up form smaller meaningful units – morphemes?
4. What are the constraints on the order of morphemes within words?
5. Do phonological rules complicate the problem of morphological analysis?

Theoretical morphologists portray word structure as being very intricate and complex, whereas computational morphologists have built working system that deal with large fragments of the morphology of languages in which the morphology is seemingly

quite complex. No existing processing system can handle the entire range of possibilities or morphological expression in natural language. While a theoretical morphologist is interested in assigning phonological elements to morphs and realizing surface realties into abstract properties, a computational morphologists, on the other hand may be interested in more immediate goals, such as providing a working system which will handle 99 percent of the words encountered in texts.

The present paper aims to address the strategies to be adopted and the linguistic and computational issues to be faced while building a tool which aims to analyze and generate the verbal complex of Tamil. Broadly speaking, the task at our hand can be divided into two:

1. Understanding and tackling of verbal complexity
2. Computational analysis and generation of the verbal complexes

2. **Understanding and tackling of verbal complexity**

The understanding of verbal complexity involves understanding the structure of simple finite verbs and compound verbs from the point of view of analysis and generation and the characteristic features of compound verbs. By understanding the nature of the verbal complexity, it is possible to evolve a methodology to tackle the verbal complexity.

2.1. Composition of a finite verb

A simple Tamil finite verbal form consists of a verbal root followed by a tense marker or negative marker and person-number-gender markers.

Verb + Tense/Negative+PNG

cey-t-aan ‘do-PAST-he’

cey-aa-tu ‘do-NEG-it’

2.2. Composition of compound verbs

A compound verb consists of a non-finite verb followed by one or more auxiliary verbs. The verbal pattern can be generalized as follows:

Verb+(Tense/Negative)+Nonfinite suffix + Auxiliary Verb....Auxiliary Verb+ Tense/Negative + PNG

cey-t-u viT-T-aan ‘do-PAST-ADVP leave-PAST-he’

‘he did it (completive sense is emphasized)’

The non-finite forms (which include certain type of verbal noun forms) of a verb can be of at least five types:

Type	**Structural description**	**Example**
1. Adverbial form	**Verb + Adverbial marker**	*cey-tu* ‘having done'
"	**Verb + Negative Adverbial marker**	*cey-aatu* ‘having not done'
2. Adjectival form	**Verb + Tense/Negative + Adjectival marker**	*cey-t-a* ‘that which is done' *ceyy-aat-a* ‘that which is not done’
3. Infinitive form	**Verb + Infinitive marker**	*ceyy-a* ‘to do’
4.1. *al*-Nominlized verbal form	**Verb+ Nominalizer** *al*	*ceyy-al* ‘doing'
4.1. *atu* - Nominlized verbal form	**Verb+Tense/ Negative+Nominalizer *atu***	*cey-t-atu* ‘that which is done'
5. Conditional form	**Verb + PAST-*aal***	*cey-t-aal* 'if done'
"	**Verb + Negative + *viT*-PAST-*aal***	*ceyy-aa-viT-T-aal* ‘if not done'

The auxiliary verbs are added to the adverbial forms, infinitive forms, *al*-nominalized forms and *atu*-nominalized forms. The flow chart appended to this article depicts the formation of all the verbal forms in Tamil. Click here for the **Flow Chart - Formation of All Verbal Forms in Tamil.**

2.3. Characteristic features of the compound verbs

The compounds show certain characteristic features that distinguish them from phrases or conjoined verbs.

1. The constituents of a verbal compound cannot be intervened by other words, whereas the conjoined verbs allow other words to intervene between them.
2. The order of the constituents in a compound is fixed. The cohesive nature of the compound will be lost if we resort to any reversal of the order or displacement of the constituents of the compound.
3. The constituents of a compound are combined into a single unit and so they are referentially opaque. It is not possible to factor out the individual constituents of a compound and apply a syntactic operation on it as the compound verbs function as simple words showing lexical integrity.
4. The compound verbs are not derived through clausal embedding, whereas conjoined verbs are derived through clausal embedding.
5. Hook (1974:94) uses grammaticalization as one of the discriminating properties of compound verbs. The same phenomenon can be adopted for Tamil and other Dravidian languages too. The V2 of compounds lose their primary lexical meaning, whereas the V2 of conjoined verbs retain their primary lexical meaning.

   Examples

   avan poo-n-aan

   he go_past_he

   ‘He went’

   avan ce-ttup poo-n-aan

   ‘he die_advp go_past_he

   ‘He passed away’

6. When negative verb *illai* is added to V1-V2 complex, it negates the entire verbal complex (i.e. the main verb),

whereas in the case of conjoined verbs it negates only the verb to which it is added.

Examples

avan ce-ttup poo-ka-villai

he die_advp go_inf_not

'He did not pass away'

avan uNavu caapiT-Tu viT-Tup paTi-kka-villai

he food eat_advp leave_advp study_not

'After having eating the meals, he did not study'

2.4. Classification of the auxiliary verbs based on their function

The following chart shows the classification of auxiliary verbs based on their function.

Type	**Semantic features**	**Constraint on the main verb**	**Verbal form**
Argument structure changing verbs			V+ infinitive suffix + *cey*
	Cause		*V+ infinitive suffix + vai*
	Cause	with a set of intranstive verbs such as*caa* 'die', *tiNaRu* 'suffocate'	V+ infinitive suffix + *aTi*
	Allow	**with intransitive verbs**	V+ infinitive suffix + *viTu*
	Allow	with intransitive verbs denoting change of state such as *kaay* 'dry', *ndanai*'become wet'	V+ infinitive suffix + *pooTu*
Voice	Passive	with transitive verbs which	V + infinitive

changing verbs		entails change of state of the object or impact on the object	suffix + *paTu*
	Passive	with transitive verbs which entails change of state of the object or impact on the object	V + infinitive suffix + *peRu*
	Reflexive	with transitive verbs denoting impact	V + adverbial suffix + *koL*
	Reciprocal	with transitive verb with plural subject	V + adverbial suffix + *koL*
	Repetitive		V+ adverbial suffix + *vaa*
Aspectual verbs	Completion Perfect		V+ adverbial suffix +*aayiRRu*
	Perfect		V+ adverbial suffix + *viTu*
	Perfect (The impact of the action is still felt.)		V+ adverbial suffix + *iru*
	Completion Perfect (Intensifies the completion and swiftness of the action.)	with intransitive verbs denoting change of state	V+ adverbial suffix + *poo*
	Completion Perfect		V+ adverbial suffix + *tiir*
	Completion Perfect (Lends force or intensity to the action denoted by the main verb.)	with transitive verbs entailing change of state of the object such as *uTai*'break', *kizi* 'tear', etc.	V+ adverbial suffix + *eRi*
	Imperfect Progressive	with verbs such as *uTkaar* 'sit' , *paTu*'lie down'	V+ adverbial suffix + *iru*
	Imperfect Progressive		V+ adverbial suffix +*koNTiru*
	Imperfect Progressive		V+ adverbial suffix +*koNTupoo*

	Imperfect Progressive		V+ adverbial suffix +*koNTuvaa*
	Imperfect Progressive (Denotes that the state described by the main verb remains undisturbed.)		V+ adverbial suffix + *kiTa*
Modal auxiliaries	Inceptive		V+ infinitive suffix + *poo*
	Attemptive (Denotes that the action expressed by the main verb is done as an attempt or trail or to see what it leads to.)		V + adverbial suffix + *paar*
	Abilitative (Denotes the capability or the potentiality on the part of the subject to perform the action denoted by the main verb.)		V + infinitive +*muTiyum*
	Abilitative (Denotes the capability or the potentiality on the part of the subject to perform the action denoted by the main verb.)		V + infinitive +*iyalum*
	Inability		V + infinitive +*muTiyaatu*
	Inability		V + infinitive +*iyalaatu*
	Possibility		V+ infinitive +*kuuTum*
	Probabilitative		V-*al* + *aam*
	Permissive		V-al + aam
	Prohibitive		V+ infinitive +*kuuTaatu*

	Prohibitive		V-*al* +*aakaatu*
	Obligative (Denotes the obligatoriness of the action denoted by the main verb)		V+ infinitive +*veeNTum*
	Dispensative (Denotes the non necessity of the action denoted by the main verb.)		V+ infinitive +*veeNTiyatillai*
	Denial		V+ infinitive +*maaTTu*
	Accelerative (Denotes the accelerated phase or rate in which the action denoted by the main verb is performed.)		V+ past participle+*taLLu*
	Denotes that the action is carried out or performed as a fore thought.		V+ past participle +*vai*
	Benefactive (Denotes that the action is carried out for the benefit of somebody.)		V + past participle +*koTu*
	Contemptive-1 (Denotes the negative attitude on the speaker towards the action expressed by the main verb.)		V + past partciple +*tolai*
	Contemptive-2 (Denotes the negative opinion of the speaker regarding the accomplishment of the action denoted by the main verb.)		V + past partciple +*kizi*
	Contemtive-3		V + past

			partciple +*azu*

2.5. Classification of auxiliary verbs based on the non-finite form to which they are added

The following chart chows the classification of auxiliary verbs based on the non-finite form to which they are added.

Type of Non-finite form to which auxiliary verbs are added	**The auxiliary verbs added after the Non-finite forms**
The auxiliary verbs added after adverbial form	aayiRRu, iru, kaaTTu, kiTa, kizi, koTu, koNTiru, koL, tiir, taLLu, tolai, paar, poo, pooTu, muTi,viTu
The auxiliary verbs added after infinitive form	aTTu, iru, iyalum, illai, kuuTum, kuuTaatu, cey, paar, poo, maaTTu, muTi, muTiyum, vaa, veeNTaam veeNTiyiru, veeNTi vaa,veeNTum
The auxiliary verbs added after *al*-nominalized form	aam, aaku, aakaatu
The auxiliary verb added after *atu*-nominalized firm	uNTu, illai, kiTaiyaatu

2.6. Multiple Natures of Verbal forms

Accounting the multiple natures of verbal forms economically is the major problem in the computational analysis and generation of verbal complex in Tamil. The simplest non-economical way is to list down all the possible combinatory forms (which runs into thousands if we take into account the addition of auxiliary verbs. But the non-economical way is against the interest of a linguist. A simple Tamil verb has a primary imperative form (which is normally the citation form for verbs), other imperative/permissive forms, finite and non-finite forms, pronominalized forms, regular verbal noun forms, etc. The following table shows the list of the paradigms of verb *cey* `to do' keeping aside the combinatory forms with auxiliary verbs.

Type of forms	**Forms of *cey***
Imperative/ Permissive forms	1. cey 2. ceyy-ungkaL 'do_you (honorific)' ='you (honorific) do' 3. ceyv-iir 'do_you' = 'You do' 4. ceyy-al-aam 'do_NOM INATIVE_may ='You may

	do’ 5. cey-ka ‘do_HORTATIVE’ = ‘Please do’ 6. ceyy-aTTum ‘do_let’ = ‘Let him/she/it do’ 7. ceyy-aatee ‘do_not’ = ‘Don’t do’
Finite forms	1. cey-t-een ‘do_PAST_I’ = ‘I did’ 2. cey-kiR-een ‘do_PRESENT_I’ = ‘I do’ 3. cey-v- een ‘do_FUTURE_I = ‘ I will do’ 4. cey-t-oom ‘do_PAST_we = ‘We did’ 5. cey-kiR-oom ‘do_PRESENT_we = ‘We do’ 6. cey-v-oom ‘do_FUTURE_we = ‘We will do’ 7. cey-t-aay ‘do_PAST_you = ‘You did’ 8. cey-kiR-aay ‘do_PRESENT_you = ‘You do’ 9. cey-v-aay ‘do_FUTURE_you = ‘You will do’ 10.cey-t-iir ‘do_PRES_you = ‘You (hon.) did’ 11. cey-kiR-iir ‘do_PRES_you = ‘You (hon.) do’ 12. cey-v-iir ‘do_FUTU_you = ‘You (hon.) do’ 13. cey-t-iirkaL ‘do_PAST_you = ‘You (hon.) do’ 14. cey-kiR-iirkaL ‘do_PAST_you = ‘You (hon.) did’ 15. cey-v-iirkaL ‘You (honorific) will do’ 16. cey-t-aan ‘do_PAST_he’ = ‘He did’ 17. cey-kiR-aan ‘do_PRESENT_he = ‘He does’ 18. cey-v-aan ‘do_FUTURE-he = ‘He will do’ 19. cey-t-aaL ‘do_PAST_she = ‘She did’ 20. cey-kiR-aaL ‘do_PRESENT_she = ‘She does’ 21. cey-v-aaL ‘do_FUTU_she = ‘She will do’ 22. cey-t-aar ‘do_PAST_he/she = ‘He/She did’ 23. cey-kiR-aar ‘do_PRESENT_he/she = ‘He/she does’ 24. cey-v-aar ‘do_FUTURE_he/she = ‘He/she will do’ 25. cey-t-aarkaL ‘do PAST_they = ‘They did’ 26. cey-kiR-aarkaL ‘do_PRESENT_they = ‘They do’ 27. cey-v-aarkaL ‘do_FUTURE_they = ‘They will do’ 28. cey-t-atu ‘do_PAST_it = ‘It will do’ 29. cey-kiR-atu ‘do_PRESENT_It = ‘it does’ 30. ceyy-um ‘do_FUTURE_it = ‘It will do’ 31. cey-t-ana ‘do_PAST_they = ‘They (neuter) did’ 32. cey-kinR-ana ‘do_PRESENT_they (n) = ‘They (neuter) do’ 33. ceyy-um ‘no_they (neuter)’ = ‘It/They (neuter) will do’
Nonfinite forms	1. cey-tu ‘do_ADVERBIAL PARTICIPLE’ 2. ceyy-aamal ‘do_NEGATIVE ADVERBIAL PARTICIPLE’ 3. ceyy-aatu ‘do_NGATIVE ADVERBIAL PARTICIPLE’ 4. ceyy-a ‘do_INFINITIVE’

	5. cey-t-a ‘do_PAST_ADJECTIVAL PARTICIPLE’ 6. cey-kiR-a ‘do_PRESENT_ ADJECTIVAL PARTICIPLE’ 7. ceyy-um ‘do_FUTURE_ ADJECTIVAL PARTICIPLE’ 8. cey-aat-a ‘do_NEGATIVE_ ADJECTIVAL PARTICIPLE’ 9. cey-t-atu ‘do_ PAST_NOMINATIVE 10. cey-kiR-atu ‘do_PRESENT_NOMINATIVE 11. cey-v-atu ‘do_FUTURE_NOMINATIVE 12. cey-aat-atu ‘do_NEGATIVE_NOMINATIVE 13. cey-t-aal ‘do_NOMINATIVE
Conditional forms	1.cey-t-aal ‘do_PAST_CONDITIONAL’
Pronominalized forms	1. cey-t-a-van ‘do_PAST_he 2. cey-kiR-a-van ‘he_PRESENT_he 3. cey-p-a-van ‘do_FUTURE_he 4. cey-aat-a-van ‘do_PAST_he 5. cey-t-a-vaL ‘do_PAST_ ADJECTIVAL _she 6. cey-kiR-a-vaL ‘do_PRESENT_ADJECTIVAL_she’ 7. cey-p-a-vaL ‘do_FUTURE_ ADJECTIVAL_she’ 8.cey-aat-a-vaL ‘do_NEGATIVE_ ADJECTIVAL _she’ 9. cey-t-a-var ‘do_PAST_he/she (honorific)’ 10. cey-kiR-a-var ‘do_PRESENT_he/she’ 11. cey-p-a-var ‘do_FUTURE_ADJECTIVAL_he/she(honorific)’ 12. cey-aat-a-var ‘do_NEGATIVE_ADJECTIVAL_he/she (honorific)’ 13. cey-t-a-varkaL ‘do_PAST_ ADJECTIVAL_they’ 14. cey-kiR-a-varkaL ‘do_PRESENT_ADJECTIVAL_they’ 15. cey-p-a-varkaL ‘do_FUTURE ADJECTIVAL_they’ 16. ceyy-aat-a-varkaL ‘do_NEGATIVE_ ADJECTIVAL_they’ 17. cey-t-a-tu ‘do_PAST_ ADJECTIVAL_it’ 18. cey-kiR-a-tu ‘do_PRESENT_ ADJECTIVAL_it’ 19. cey-v-a-tu ‘do_FUTURE_ ADJECTIVAL_it’ 20. ceyy-aat-a-tu ‘do_NEGATIVE_ADJECTIVAL_it’ 21. cey-t-a-na ‘do_PAST_ ADJECTIVAL_they’ 22. cey-kinR-ana ‘do_PRESEENT_ ADJECTIVAL_they’ 23. cey-v-a-na ‘do_FUTURE_ ADJECTIVAL_they’ 24. ceyy-aat-a-na ‘do_NEGATIVE_ ADJECTIVAL_they’
Regular verbal noun forms	1. ceyy-al ‘do_NOMINATIVE’ 2. cey-tal ‘do_NOMINATIVE’ 3. cey-kai ‘do_NOMINATIVE’

	4. ceyy-aa-mai ‘do_NOMINATIVE’

If we calculate the possible forms of each verb it will be as follows:

Imperative/Permissive forms - 7
Finite forms - 33
Non-finite forms - 11
Conditional form - 1
Pronominalized forms - 24
Regular Verbal noun forms - 4
Total - 80

Thus we will have at least 80 paradigms to be accounted for each verb. This number will multiply if we take into account the compound verb formation by adding auxiliary verbs with the main verbs. Storing all these forms for the 3012 verbs listed in KTTA (*Kiriyaavin taRkaalati tamiz̲ akaraati*) in the memory of computer is a waste. And more over we need to generate these forms from verb stems and suffixes. As the tense inflection is not regular and is not phonologically conditioned, we need to classify the verbs into certain classes to capture the inflectional process.

2.7. Tackling of the verbal complexity

In order to tackle the analysis and generation of the verbal forms in which the inflection vary from one set of verbs to another, a classification of Tamil verbs based on tense inflections need to be evolved. The inflection includes finite, infinite, adjectival, adverbial and conditional forms of verbs. For the sake of computation we have to make twentyeight classes of verbs and group the verbs under each group. For the sake of convenience of remembering them along with the morphophonemic changes involved in the inflectional and derivational processes, we can label them as *cey*-class, *azu*-class, *caa*-class, *viTu* class, *peRu*-class, *pukaz*-class, *vizu*-class, *aRi*-class, *ndoo*-class, *vee*-class, *vaa*-class, *akal*-class, *kol*-class, *aaL*-class, *koL*-class, *ooTu*-class, *col*-class,*poo*-class, *kal*-class, *keeL*-class, *puuN*-class, *uN*-calss, *kaaN*-class, *tin*-class, *en*-class, *ndil*-class, *paTi*-class, and *ndaTa*-class.

The following morphophonemic information on these classes of verbs will show the need for classifying the verbs into 28 classes.

(The preference is given here for the item arrangement model rather that item and process model.)

1. cey-class: The verbs of the cey class take -t- as past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense maker and tu as the adverbial participle marker. The final consonant of the verbs geminates when the suffixes beginning with a vowel such as um, a, aat, al are added to them.

2. azu-class: The verbs of azu-class take -t- as past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense marker and tu as the adverbial participle marker. The final u of the verbs is deleted when the suffixes beginning with vowel such as um, a, aat, al are added to them.

3. caa-class: The verb of caa-class takes t as past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense marker and ttu as the adverbial participle marker. The irregular verb caa is the only verb belonging to this class. caa changes into ce before past tense marker t and adverbial marker ttu. An increment k is added in between when the suffixes beginning with vowel such as um, a, aata, al are added to the verb stem directly.

4. viTu-class: The verbs of viTu-class take T (i.e. Tu+t > TT) as the past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense marker and Tu (i.e. Tu+tu > TTu) as the adverbial participle marker. The final vowel u of the verbs is dropped when past tense suffix and the adverbial suffix as well as the suffixes beginning with vowel such as um, a, aat, al are added to them.

5. peRu-class: The verbs of peRu class take R (i.e. Ru+t > RR) as the past tense marker, kiR ~ kinR as present tense marker, v ~ um as the future tense marker and Ru (i.e. Ru+tu > RRu) as the adverbial participle marker. The final vowel u is dropped when past tense suffix and the adverbial suffix as well as the suffixes beginning with vowel such as um, a, aat, al are added to the verb stem.

6. pukaz-class: The verbs of pukaz-class take ndt as the past tense marker and kiR~kinR as present tense marker, v ~ um as the future tense marker and ndtu as the adverbial participle marker.

No change in the phonemic make up of the stems is effected when the verbs of this type inflect.

7. vizu-class: The verbs of vizu class take ndt as the past tense marker, kiR ~ kinR as present tense marker, v ~ um as the future tense marker and ndtu as the adverbial participle marker. The final u of the verb is dropped, when the suffixes beginning with vowel such as um, a, aat, al are added to the verb directly.

8. aRi-class: The verbs of aRi-class take ndt as the past tense marker, kiR~kinR as present tense marker, v ~ um as the future tense marker and ndtu as the adverbial participle marker. An onglide y is inserted when the suffixes beginning with vowel such as um, a, aat, al are added directly to the verbs.

9. ndoo-class: The verbs of ndoo class take ndt as the past tense marker and kiR~kinR as present tense marker, v ~ um as the future tense marker, and ndtu as the adverbial participle marker. The verb stem ndoo changes into ndo when past tense marker and adverbial participle marker are added to the stem. When the suffixes beginning with vowel such as um, a, aat, al are added to the verb stem directly, an increment k is added in between.

10. vee-class: The verbs of vee-class take ndt as the past tense marker and kiR~kinR as present tense marker, v ~ um as the future tense marker, and ndtu as the adverbial participle marker. The verb stem vee changes into ve when past tense marker and adverbial participle marker are added to the stem. When the suffixes beginning with vowel such as um, a, aat, al are added to the verb stem directly, an increment k is added in between.

11. vaa-class: (varu is taken as the base form rather than the citation form vaa). The verbs of vaa-class take ndt as the past tense marker, kiR~kinR as present tense marker, and v ~ um as the future tense marker. u is dropped when the set of suffixes beginning with vowel such as um, a, aat, al are added directly to the verb stem. varu changes into va before past tense marker and adverbial participle marker and vaa before imperative marker. vaa (varu~vaa~va) and taa (~taa~ta) are the two members of this class.

12. akal-class: The verbs of akal class take nR (i.e. l+ndt > nR) as the past tense marker and kiR~kinR as present tense marker, v ~ um as the future tense marker and nRu (l+ndtu > nRu) as the adverbial participle marker. The final l of the verbs becomes nil when it assimilates with the past tense suffix and adverbial suffix.

13. kol-class: The verbs of kol class take nR (l+ndt > nR) as the past tense marker, kiR~kinR as present tense marker, v ~ um as the future tense marker, and nRu (l+ndtu > nRu) as the adverbial participle marker. The final l becomes nil when it assimilates with the past tense suffix and adverbial suffix. The final l of the verbs geminates into ll when the set of suffixes beginning with vowel such as um, a, aat, al are added directly to the verbs.

14. aaL-class: The verbs of aaL class take NT (i.e. L+ndt > NT) as the past tense marker, kiR~kinR as present tense marker, v ~ um as the future tense marker, and NTu (i.e. L+ndtu > NTu) as the adverbial participle marker. The final L of the verbs becomes nil when it assimilates with the past tense suffix and adverbial suffix.

15. koL-class: The verbs of koL class take ndt (i.e. L+ndt > NT) L assimilates with the past tense suffix.) as the past tense marker, kiR~kinR as present tense marker, v ~ um as the future tense marker, and NTu (i.e. L+ndtu > NTu) as the adverbial participle marker. The final L becomes nil when it assimilates with the past tense suffix and adverbial suffix. The final L of the verbs geminates into LL when the set of suffixes beginning with vowel such as um, a, aat, al are added directly to the verbs.

16. ooTu-class: The verbs of ooTu class takes in as past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense marker and i as the adverbial participle marker. The final u of the verbs is deleted when the suffixes beginning with vowel such as um, a, aat, al are added to the verbs directly.

17. col-class: The irregular verb col is the only member of this group. The verb of col-class takes nn (l+in > nn) as past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense marker, and i (in > i) as the adverbial participle marker. The final l becomes nil when it assimilates with the past tense suffix. The final l of the verbs geminates into ll when the set suffixes

beginning with vowel such as um, a, aat, al, i are added directly to the verbs.

18. poo-class: The verb of poo-class takes n (in > n) as the past tense marker, kiR ~ kinR as present tense marker, v ~ um as future tense marker, and y (in > y) as the adverbial participle marker. An increment k is added in between when the suffixes beginning with vowel such as um, a, aat, al are added to the verb stem directly. The verbs poo and aa are the two members of this class.

19. kal-class: The verbs of kal class take R (l+t > RR) as the past tense marker, kiR~kinR as present tense marker, p ~ um as the future tense marker, and Ru (l+tu > RRu) as the adverbial participle marker. The final l of the verbs changes into R when the suffix beginning with consonant such as the past tense suffix R, adverbial suffix Ru, present tense suffix kiR ~ kinR, future tense suffix p are added to the verbs directly. An increment k is added in between when the suffixes beginning with vowel such as um, a, aata, al are added to the verbs directly. The final l of the verbs changes into R even before the increment k.

20. keeL-class: The verbs of keeL class take T (L+t > TT) as the past tense marker and kiR ~ kinR as present tense marker, p ~ um as the future tense marker and Tu (L+tu > TTu) as the adverbial participle marker. The final L of the verbs changes into T when the suffixes beginning with consonant such as the past tense suffix T, adverbial suffix Tu, present tense suffix kiR ~ kinR, future tense suffix p are added to the verbs directly. An increment k is added in between when the suffixes beginning with vowel such as um, a, aata, al are added to the verbs directly. The final L of the verbs changes into T even before the increment k.

21. uN-class: The verbs of uN-class take T (i.e. N+t > NT) as the past tense marker, kiR ~kinR as present tense marker, p ~ um as the future tense marker and Tu as the adverbial participle marker. The final N of the verbs geminates into NN when the suffixes beginning with vowel such as um, a, aat, al are added to the verb stem directly.

22. puuN-class: The verbs of puuN-class take T (i.e. N+t > NT) as the past tense marker and kiR~kinR as present tense marker and p ~ um as the future tense marker, and NTu (i.e. N+tu > NTu) as the adverbial participle marker.

23. kaaN-class: kaaN is the only member of this group. The verb of kaaN-class takes T (i.e. N+t > NT) as the past tense marker and kiR~kinR as present tense marker and p ~ um as the future tense marker and NTu (i.e. N+tu > NTu) as the adverbial participle marker.

24. tin-class: The verbs of tin class take R (i.e. n+t > nR) as the past tense marker and kiR~kinR as present tense marker, p ~ um as the future tense marker and Ru (i.e. n+tu > nRu) as the adverbial participle marker. The final n of the verbs geminates into nn when the suffixes beginning with vowel such as um, a, aata, al are added to the verbs directly.

25. en-class: The verbs of en-class take R (i.e. n+t > nR) as the past tense marker and kiR~kinR as present tense marker, p ~ um as the future tense marker and Ru (i.e. n+tu > nRu) as the adverbial participle marker. Unlike the previous class the final n of the verbs does not geminate into nn when the suffixes beginning with vowel such as um, a, aata, al are added to the verbs directly.

26. ndil-class: ndil is the only member of this class. The verb of ndil class take nR (i.e. l+ndt > nR) as the past tense marker, kiR~kinR as present tense marker, p ~ um as the future tense marker and nRu as the adverbial participle marker. The final l becomes nil when it assimilates with the past tense suffix nR and adverbial suffix nRu. The final l of the verbs changes into R when the suffixes beginning with consonant such as present tense suffix kiR ~ kinR, future tense suffix p are added to the verbs directly. An increment k is added in between when the suffixes beginning with vowel such as um, a, aat, al are added to the verbs directly. The final l of the verbs changes into R even before the increment k.

27. paTi-class: The verbs of paTi class take tt as the past tense marker, kkiR~kkinR as present tense marker, pp ~ um as the future tense marker and ttu as the adverbial participle marker. An

increment kk is added in between when the suffixes beginning with vowel such as um, a, aat, al are added to the verb stem directly.

28. ndaTa-class: The verb of ndaTa class take ndt as the past tense marker and kkiR ~ kkinR as present tense marker, pp ~ um as the future tense marker and ndu as the adverbial participle marker. An increment kk is added in between when the suffixes beginning with vowel such as um, a, aata, al are added to the verb stem directly.

2.7. TAM chart

The flow chart given above depicts the analysis to be followed. Unlike the regular morphological analysis which is restricted within words and whose function to strip off the affixes by assigning them meaning and identifying the lexeme to be matched with the item in the dictionary, the analysis of verbal complex goes beyond words as the verbal complex contains affixes as well as words which are stringed into a unit. So a TAM chart is prepared showing the morphemes (which includes suffixes as wells as auxiliary verbs) to be added to each verb at the horizontal axis and the verb classes at the vertical axis. (The TAM chart is divided into two parts to adjust with the space.) Click here for the **TAM CHART.** The following **eight charts** are given in the above link:

Table 1. The vebal bases occurring at the first level and the suffixes occurring at the second level.
Table 2: The suffixes occurring at the third and fourth level.
Table 3: The third level auxiliary verbs occurring after infinitive and the suffixes of second occurring at the fourth level.
Table 4: The third level auxiliary verbs occurring after infinitive and the suffixes of third level occurring at the fifth level and the suffixes of fourth level occurring at the sixth level.
Table 5: The third level auxiliary verbs occurring after adverbial participle and the suffixes of second level occurring at the fourth level.
Table 6: The third level auxiliary verbs occurring after adverbial participle and the suffixes of third level occurring

at the fifth level and the suffixes of fourth level occurring at the sixth level.
Table 7: The auxiliary verbs occuring after al-nominalized form.
Table 8: The auxiliary verb occuring after atu-nominalized firm.

3. Computational analysis and generation of the verbal complex

Computational analysis of verbal complex involves two processes:

1. The analysis of verbal complex into minimal meaningful units (which includes suffixes and bound auxiliary verbs) and assigning semantic features to each unit, and
2. The generation of verbal complex out of the same meaningful units.

Let us make an assumption that the building of verbal complex is the concatenation of morphemes and bound words. Clearly the simplest way to build such a model would be to assume that the allowability of a particular morpheme or a bound word in a given context depends only upon the morpheme that precedes it. Take the word ceytuviTTeen, whose morphological structure is cey+tu + viT+T+een. The combination of cey which is a verb and tu which an adverbial marker forms the first chunk of the verbal complex and the combination of viT which is a secondary verb, T which is a past tense marker and een which is a person-number-marker forms the second chunk of the word. Both of the chunks (V1 and V2) make a verbal complex.

Finite State Automata

One can easily model such a system with a finite-state machine (finite-state automaton or finite-state transition network) of the kind familiar from formal language theory. More formally, a finite automation is defined by the following five parameters:

Q: a finite set of N states q0, q1..., qN
Σ: a finite input alphabet of symbols
q0: the start state

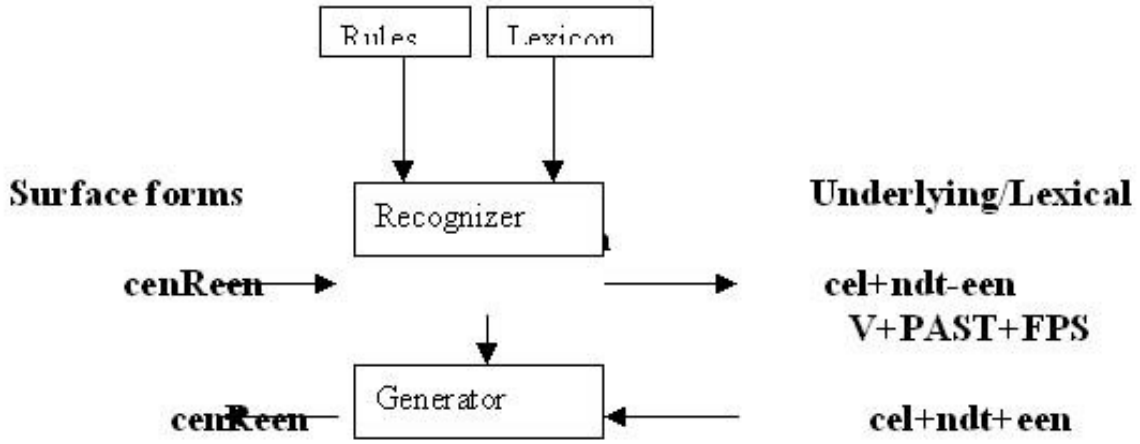
F: the set of final states, F subset of or equal to Q
δ (q,i): the transition function or transition matrix between states. Given a state q element of Q and an input symbol i element of Σ δ (q,i) returns a new state q' element of Q. δ is thus a relation from Q x Σ to Q;

A set of rewrite rules, or grammar, can be thought of not only as an intentional definition of the units of a language, but also as the description of an abstract machine which is able to accept (or generate) just those strings which belong to the language. These machines are called (abstract) automata, and the complexity of their potential behaviour is directly determined by the internal complexity of the rules of the grammar to which they correspond. All such machines may be characterized in terms of a set of states, an input device which can access one input symbol at a time, and control unit which can examine and read input and cause the machine to shift from one sate to another. In addition, such machines may have memory, which can be accessed by the control unit for storing and testing symbols. The computational power of the machine is determined essentially by the complexity of its memory and of the operations that can be performed on the memory. The simplest of these automata are called finite state machines or finite state automata (FSA), and the languages they can accept are regular languages.

PCKimmmo

The two level morphology of Kosekenniemi has been a widely implemented one, particularly in European languages. PCKimmo can do two functions as a morphological processor, recognition. In the case of the generator component of PCKimmo, it takes a lexical form of the foot as input. To this imput rules recorded in the rules file are applied and the corresponding surface surface form is returned. It does not use the lexicon. The recognizer accepts as input a surface form, applies the rules, consults the lexicon and returns the corresponding lexical form and gloss string. What makes it easier for the user and important from the point of economy of model is the fact that the same set of rules can be used for both the functions of recognition and gneration. The rules can be used birectionally (1990:8). If certain phonological, or of for that matter a morphological or orthographic, rule is written to refer to a

transformation form the underlying structure to surface structure, they need not be rewritten for the reverse process. In reverse, it processes form the surface form to the underlying form. This model can be represented through the following diagram (Simons 1989).



(FPS=First person singular)

The two levels used in this model are linguistic descriptions that can handle any tow levels of abstractions, be it allophonic alternations, morphophonemic alternations or orthographic variations.

API Morphological Analyzer

The API Morphological analyzer developed by AUKBC Research Centre in collaboratin with the first author of the paper depends on automata table present in its data file, statetable.dat. The API morphological Analyzer is actually designed to strip a word of its inflections.

The machine while processing is set to be at the state beging represented by 0:0. Then the input is looked into root dictionary for a match. If a match is found it returns it along with its gloss and by printing, ‘FINISH’ in the command line. If it doesn’t find the input in the root file, then it starts matching characters form the right preriphery with the entries listed in the file ‘statetable.dat.’ If it doesn’t find a match there then it goes and checks in a file called ‘SANDHI.dat’. If it finds a morpheme, then it strips the word of the

morpheme, takes the remaining part of the word to be checked again in a file called ‘ROOT.dat’. The sandhi replacement tables are in the file ‘sandhidict.dat’. For all these operations the analyzer depends on automata tables present in the datafile ‘statetable.dat’. This table contains various rules defining the paths needed to identify the affixes in a Tamil word. The state table is based on the verb flow chart. The format in which the rules are represented in this file is as fallows:

Stateno:morpheme,next state [-nextstate...]; [morpheme, nextstate];

An example for a morpheme in the suffix file will be as follows:

Ex.3.0:avan,1.1;avaL,1.1; avar,1.1;atu,1.1;avai,1.1; ana,1.1; kai,2.1.;tal,2.1; #ttal,2.1-End;Ral,2.1.-End

The fist entry in the line (stateno) is an arbitray number (or word) designating the state that the automata must be in when it scans the line. This ‘stateno’ is separated form other entries by a colon. Any number of entries may follow the ‘stateno’. A semicolon terminates each entry. The semicolon is only optional for the last entry. Each entry of two fields, the morpheme and the number of the next state that the automata will transit into if the morpheme in the entry is present at the end of word. These two fields are separated by a coma.

The following is an example for a rule in the file which strips the word the change because of the sandhi operations.

Ex. maram+ai==> marattai

The correspondence shows all that m will be replaced by tt when ai is added. During analysis tt will be replaced by the original m deletion. The entry in the rule file will be in the following format.

Ex. 4a +/- <replacement character> {{Suffix{+postfix: [postfix...]}; {Suffix{+postfix:[postfix...]};

The morpheme entry will be as follows:

Ex.4b. –m, {tt{+ai:+aal:+uTan:+in:+ttin:+il:}; ng{+kaL:}}

The ‘-‘ sign before m indicates that the character will be added. The suffixes are enclosed in curly brackets. In the case of multiple suffixes, as is the case above, semicolons separate each suffix.

Different modules are used to do different functions. The modules have been written in the compiled languge Java. The main functions load is done by TamilMorph.java. A module called RootDict.java does the search for the match in the root file. Similarly Suffix.java, Sandhi.java etc. take care of various functions using the finite state automata.

4. Conclusion

The preparation of morphological analyzer is the preliminary step to be made before venturing into any other type of natural language processing. Tamil being a complex inflectional language is a challenge for computational linguists who try to prepare a morphological analyzer for it. There are attempts to prepare morphological generators and analyzers for Tamil. Preparing a morphological generator or analyzer for the purpose of demonstration is quite different from preparing an exhaustive and efficient working system. It is not too difficult to prepare a system with the coverage of 60 to 65%. However increasing the efficiency beyond 65% and to reach a realistic level of 95% to 97% is a hard task. Most of the existing descriptions focus on a narrow and vertical detail rather than on breadth and exhaustiveness. Generally the coverage obtained by incorporating the descriptions of the existing work is never beyond 50%. The evaluation method points out that the efficiency of the morphological analyzer prepared for Tamil by the present team is very remarkable.

The preparation of a morphological analyzer, in which the processing of verbal complex forms a part, has may natural language applications such as parsing, text generation, machine translation, preparing dictionary tools and lemmatization. It also helps in speech applications such as text-to-speech synthesizing and speech recognition and in word processing applications like spell checking and text input and in retrieval of documents. Finding the categorical details of the verbal forms helps in assigning parts-of-speech tags to the verbal complex.

*** *** ***

Acknowledgement

Our thanks are due to the other members of the NLP team of Anna University KBC Research Centre, Mr.B.Kumarashanmugam, Mr.S.Baskaran and Mr.S. Arulmozi and also to Prof. Vijaya Krishnan and Dr.Ramanan who have shaped it up to the present state.

REFERENCES

Agesthialingom, S. 1964. 'Auxiliary verbs in Tamil'. ***Tamil Culture*** 11:3.

Annamalai,E. 1985. ***Dynamics of verbal extension***. Trivandrum: Dravidian Linguistics Association.

Dale, I.R.H. 1980. 'Syntax, Semantics, and Tamil auxiliary verbs'. In: ***Auxiliaries in Dravidian***. (eds.). S.Agestialingom & G. Srinivasavarma. Annamalainagar: Annamalai University.

Hook, P.E. 1974. **The compound verb in Hindi**. Ann Arbor: University of Michigan.

Joseph, N. 1983. ***Auxiliaries in Tamil***. Ph.D Thesis. Annamalainagar: Annamalai University.

Jurafsky, D and Matin, J.H. 2000. ***Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition***. New Jersey: Prentice Hall.

Lehman, T. 1993. ***A grammar of Modern Tamil***. Pondicherry: Pondicherry Institute of Linguistics and Culture.

Rajendran, S. 1999. ‘Spell and grammar checker for Tamil’. Paper read in 27th All India Conference of Dravidian Linguists held in ISDL, Thiruvananthapuram, 17th-19th, July, 1999.

----------. 2001. ***Wordfomation in Tamil*** (manuscript). Thanjavur: Tamil University.

Rajendran, S., Arulmozi, S., Ramesh Kumar, Viswanathan, S. 2001. ‘Computational morphology of verbal complex’. Paper read in Conference at Dravidan University, Kuppam, December 26-29, 2001.

Rangan, K. 1970. ‘Modals as main verbs in Tamil’. In Proceedings of the First All India Conference of Linguists. Poona: Deccan College.

Ranganathan, Vasu. 1997. ‘A Lexical Phonological Approach to Tamil Word by computer’. International Journal of Dravidian Linguistics 26.1:57-70.

Rao, Umamaheshwar G. 1996. ‘Compound verb formation in Telugu’. Paper presented in the National Workshop-cum-seminar on Lexical Typololgy, Telugu University, Hyderabad.

Roper, T. and Siegel, D. 1978. ‘A lexical transformation for verbal compounds’. ***Linguistic Inquiry*** 9.199-260.

Sengupta, G. 1997. ‘Three Models of Morphological Processing.’ ***South Asian Language Review***, 8:1, 1-26.

----------. 1998. ‘Language Technology and Morphological Processing’. ***The Administrator***, 43:3.

Simons, G. 1989. ‘A tool for exploring morphological’ ***Notes on Linguistics*** 44:51-59.

Singh, J. 1994. ‘Compound verb in Hindi: A GB approach’. Paper presented in National seminar on Word formation in Indian Languages, Hyderabad, Osmania University, February 9-10, 1994.

Sproat, R. 1992. ***Morphology and computation***. Cambridge: The MIT Press.

Steever, S.B. 1983. ***A study in auxiliation: the grammar of the indicative auxiliary verb system of Tamil***. Unpublished Ph.D. dissertation. Chicago: University of Chicago.

----------. 1993. ***Analysis to synthesis: the development of complex verb morphology in Dravidian languages.***New York: Oxford University Press.